

Optimal Control of Propagating Fronts by Using Level Set Methods and Neural Approximations

Angelo Alessandri, *Senior Member, IEEE*, Patrizia Bagnerini, Mauro Gaggero

Abstract—We address the optimal control of level sets associated with the solution of normal flow equations. The problem consists in finding the normal velocity to the front described by a certain level set in such a way to minimize a given cost functional. First, the considered problem is shown to admit a solution on a suitable space of functions. Then, since in general it is difficult to solve it analytically, an approximation scheme that relies on the extended Ritz method is proposed to find suboptimal solutions. Specifically, the control law is forced to take on a neural structure depending nonlinearly on a finite number of parameters to be tuned, i.e., the neural weights. The selection of the optimal weights is performed with two different approaches. The first one employs classical line-search descent methods, while the second one is based on a quasi-Newton optimization that can be regarded as a neural learning based on the extended Kalman filter. If compared to line-search methods, such an approach reveals to be successful with a reduced computational effort and an increased robustness with respect to the trapping into local minima, as confirmed by simulations in both two and three dimensions.

Index Terms—level set methods, normal flow, optimal control, extended Ritz method, adjoint equation, neural approximation, extended Kalman filter.

I. INTRODUCTION

Level set (LS) methods are numerical algorithms used to solve Hamilton-Jacobi equations, a particular class of first-order hyperbolic partial differential equations (PDEs) [1]. They are widely used to simulate the motion of fronts in two or three dimensions in many different fields, such as computational fluid dynamics, fluid-structure interaction, image processing, detonation or deflagration waves, seismic analysis, and materials science [2]–[6]. The various LS methods depend on the particular velocity field in the Hamilton-Jacobi equation. Among them, the most known are the normal flow equation, where the propagation speed is directed towards the normal to the front, and the mean curvature flow equation, where the speed is proportional to the curvature of the front in all the points.

The LS of a function can be considered as a front or an interface separating two regions, either a curve in two dimensions or a surface in three dimensions [1], [7]. As novel contribution w.r.t. the state of the art, in this paper we attack the problem of optimally driving a moving front described by the LSs of the solution of a normal flow equation. Based on

the preliminary results of [8], [9], we propose an approach that overcomes the computational difficulties that have prevented to face this problem up to now. Such a breakthrough is achieved by means of a quasi-Newton optimization technique, which can be regarded as a learning powered by an extended Kalman filter (EKF). The use of the EKF is motivated by its efficiency in dealing with large amounts of data [10].

Moving interfaces are a fundamental modeling tool in various research fields. In fact, many problems are characterized by a number of different regions interacting and depending on various factors, such as physical laws and geometry. A lot of examples exist in cross-disciplinary contexts, such as fluid dynamics, materials science, computational biology, biomedicine, land protection, marine and energy engineering. Several techniques are available in the literature to study the evolution of moving interfaces. They can be catalogued as front tracking and front capturing methods. The former ones are Lagrangian, i.e., the front is discretized using a mesh, while the latter ones are Eulerian, i.e., the interface is represented implicitly on a fixed grid. In Lagrangian methods, a given number of points is positioned along the front and then moved using a system of ordinary differential equations [11], [12]. Such methods are very efficient and accurate in the case of interfaces characterized by small deformations, but they may be quite difficult to be used with changes of topology, and remeshing may be needed in the presence of large deformations of the boundary. Alternative approaches that makes it possible to overcome these drawbacks are given by Eulerian methods, and in particular LS methods, where the front is implicitly represented at each time by a LS (for instance, the zero level) of a multidimensional function [2]. They have various advantages over Lagrangian approaches. First, they rely on typical geometric quantities that can be easily computed, such as the curvature or the normal to the front. Then, changes of topology can be considered in an easy way. Lastly, the extension to dimensions higher than two is straightforward.

LS methods are popular in topology optimization (see, e.g., [13]). In this case, the velocity field is given by the opposite of the shape derivative, and the empty region of the domain is replaced by a weak phase to avoid singularities and extend the shape derivative to the whole domain. Unlike our approach, where the aim is to control the front evolution dynamically, shape optimization is purely static and any behavior over time is not considered, i.e., the shape does not change with time.

A huge literature concerning the control of systems described by PDEs exists, but very few contributions are available on the control of fronts described by LS methods. In fact, most of the available studies focus only on simulating

A. Alessandri and P. Bagnerini are with the University of Genoa, Via all'Opera Pia 15, 16145 Genoa, Italy (e-mail: alessandri@dime.unige.it; bagnerini@dime.unige.it).

M. Gaggero is with the National Research Council of Italy, Via De Marini 6, 16149 Genoa, Italy (e-mail: mauro.gaggero@cnr.it).

This work was supported in part by the AFOSR with grant no. FA9550-15-1-0530.

and tracking the evolution of interfaces, while only few works are available on the control of moving fronts. This may be ascribed to the theoretical and numerical difficulties one may encounter in attacking the problem, and the poor recognition of the potential application, which only in very recent time has emerged to some extent. Among the few available results, [14] presents a prey-predator model based on biology. The control of LSs resulting from the two-phase Stefan problem is the topic addressed in [15] and [16], where the solution is searched for numerically by using gradient-based methods.

As novel contribution w.r.t. the literature, in this paper we address the optimal control of LSs generated by the normal flow equation with the velocity field regarded as a control action. First, a theoretical investigation of the properties of such a problem is presented. Then, since it is almost impossible to find an analytic solution, we focus on finite-dimensional approximations based on the extended Ritz method (ERIM). Such an approach was proposed in the past to solve optimal control problems for nonlinear discrete-time systems [17]–[21]. Its basic idea is to constrain the control policy to assume a fixed, neural structure with a finite number of free parameters to be properly tuned. The original functional optimization problem is converted into a mathematical programming one that requires the optimization of the parameters. The use of the ERIM for controlling distributed parameter systems is presented in [22], showing that it can be used for the optimal control of generic PDE-based systems, as it lies in the middle between the two typical paradigms “discretize-then-optimize” and “optimize-then-discretize” [23].

In this paper, two different approaches are investigated to search for the optimal parameters of the control laws. The first one relies on classical line-search descent methods. The second one is based on an optimization derived from the Newton method. In practice, the selection of the weights is accomplished by means of an EKF learning procedure [24]–[28]. Both approaches require to compute the gradient of the cost w.r.t. the weights. Such a gradient is determined by using adjoint methods [29], [30]. As regards the control of LSs, in [9] it is shown that the use of the gradient computed by solving the related adjoint equation enables to reduce the overall simulation times as compared with its finite-difference approximation. The combination of EKF-based optimization and efficient computation of the gradient provide an increased robustness w.r.t. the trapping into local minima, in line with the preliminary results reported in [31].

This paper is structured as follows. The considered optimal control problem of moving fronts is formulated in Section II, where we define the functional space on which a solution exists. In Section III, the approach based on the ERIM is showcased, together with the adjoint equation for computing the gradient of the cost, to find approximate solutions to the optimal control problem that depend on weights to be tuned. Section IV describes the proposed methods for the optimization of the weights. Lastly, Section V presents the simulation results, while conclusions are discussed in Section VI.

We will adopt the following notation. For any column vector $x \in \mathbb{R}^n$, let $|x| := (x^\top x)^{1/2}$ denote its Euclidean norm.

Moreover, let $(x, y) := [x^\top, y^\top]^\top$, where $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$. Let \mathcal{X} be a real linear normed space of functions with the norm $|\cdot|_{\mathcal{X}}$. The functional $x \mapsto G(x) : \mathcal{X} \rightarrow \mathbb{R}$ is said to be Fréchet differentiable in $x \in \mathcal{X}$ if there exists a functional $h \mapsto G'(x)h : \mathcal{X} \rightarrow \mathbb{R}$ such that

$$G(x+h) = G(x) + G'(x)h + r(x, h)$$

where $G'(x)$ is the Fréchet derivative of $G(x)$ in $x \in \mathcal{X}$, and $h \mapsto r(x, h)$ is a remainder of order higher than one, i.e.,

$$\lim_{h \rightarrow 0} \frac{|r(x, h)|}{|h|_{\mathcal{X}}} = 0.$$

Given $A \subset \mathbb{R}^n$, the quantities \overline{A} , ∂A , and $N(A)$ denote the closure, the boundary, and a neighborhood of A , respectively. For $p \in [1, +\infty)$, a function $f : A \rightarrow \mathbb{R}^n$ belongs to $L^p(A)$ if its L^p norm is bounded, i.e., $|f|_p := (\int_A |f(x)|^p dx)^{1/p} < \infty$. Moreover, $f \in L^\infty(A)$ if $|f|_\infty := \text{ess sup}_{x \in A} |f(x)| < \infty$.

II. OPTIMAL CONTROL OF LEVEL SETS

Let us consider a set $\Omega \subset \mathbb{R}^q$ open, bounded, and smooth and the time $t \in [0, T]$, with $T > 0$. LS methods represent a moving front or interface at each time t , i.e., a curve in two dimensions or a surface in three dimensions separating two regions, as the zero LS of a multidimensional function $\phi : \overline{\Omega} \times [0, T] \rightarrow \mathbb{R}$. The interface $x(t, s)$ is given at time t by the points such that $\phi(x(t, s), t) = 0$, where s is the arc-length parameter of the initial curve $x(0, s)$. Figure 1 displays fronts at two different time instants t_1 and t_2 . By differentiating w.r.t. t , we obtain

$$\phi_t(x, t) + v(x, t) \cdot \nabla \phi(x, t) = 0 \quad (1)$$

i.e., a Hamilton-Jacobi equation, where $v(x, t) := \frac{d}{dt}x(t, s)$ is the Lagrangian particle velocity giving the direction of propagation of the interface at the point $x(t, s)$, while $\nabla \phi(x, t)$ is the Fréchet gradient of $\phi(x, t)$ w.r.t. the space. From now on, we focus on the normal flow equation, which corresponds to choose $v(x, t)$ proportional to the normal to the front, i.e.,

$$v(x, t) = u \frac{\nabla \phi(x, t)}{|\nabla \phi(x, t)|} \quad (2)$$

where u is the speed of propagation. By replacing (2) in (1), we get

$$\phi_t(x, t) + u(x, t) |\nabla \phi(x, t)| = 0 \quad (3)$$

where the speed function $u : \overline{\Omega} \times [0, T] \rightarrow \mathbb{R}$ is regarded as a control input. Equation (3) has proper initial conditions $\phi_0 : \overline{\Omega} \rightarrow \mathbb{R}$, i.e., $\phi(x, 0) = \phi_0(x)$, $x \in \overline{\Omega}$. Usually, ϕ_0 is chosen as the signed distance to the initial front. Equation (3) is a Hamilton-Jacobi equation, whose solution is defined in the sense of viscosity solutions and it is based on the notion of sub- and super-differentials [32]. The l LS of the function ϕ is a set-valued mapping $\Gamma_l : [0, T] \rightrightarrows \mathcal{C}$, where $\Gamma_l(t) := \{x \in \Omega : \phi(x, t) = l\}$.

We deal with the problem of the optimal control of (3) for some cost functional to be minimized that provides a performance index depending on the propagating front associated with a certain LS of $\phi(x, t)$. Let us denote by \mathcal{U} the set of admissible control functions $(x, t) \mapsto u(x, t)$ and by \mathcal{F}

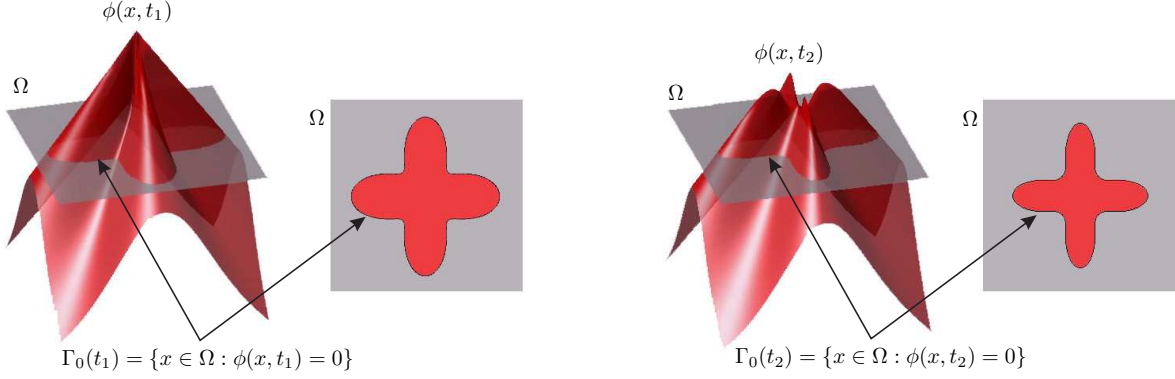


Fig. 1. Fronts described by the zero LSs of a multidimensional function ϕ at two different time instants t_1 and t_2 .

the space of functions $(t, x) \mapsto \phi(x, t)$ where the problem is formulated. In the following, we properly define \mathcal{U} and \mathcal{F} for the optimal control problem

$$\inf_{u \in \mathcal{U}, \phi \in \mathcal{F}: (3) \text{ holds}} J(u, \phi) \quad (4)$$

where $J : \mathcal{U} \times \mathcal{F} \rightarrow \mathbb{R}$ is a smooth cost functional.

First of all, let $A \subset \mathbb{R}^n$ be open and define

$$[u]_1 := \sup_{x, y \in A, x \neq y} \frac{|u(x) - u(y)|}{|x - y|}$$

where $u : A \rightarrow \mathbb{R}$. We denote the class of continuous functions and bounded continuous functions in A by $C^0(A)$ and $C_b^0(A)$, respectively. Moreover, let

$$C^{0,1}(A) := \{u \in C_b^0(A) : [u]_1 < \infty\}.$$

Then, $(C^{0,1}(A), \|\cdot\|_1)$ is complete, where $\|u\|_1 := \|u\|_\infty + [u]_1$. In other words, $C^{0,1}(A)$ endowed with the norm $\|\cdot\|_1$ is a Banach space (see, e.g., [33]). Notice that, if $[u]_1 < \infty$, then $[u]_1$ is the smallest constant L such that $|u(x) - u(y)| \leq L|x - y|$ for all $x, y \in A$, i.e., u is uniformly Lipschitz. Moreover, since a Lipschitz function is uniformly continuous and therefore continuously extendable to the boundary of its domain, it follows that

$$C^{0,1}(A) = C^{0,1}(\bar{A}).$$

Based on the aforesaid, the following propositions hold [34].

Proposition 1: If A is bounded, the immersion $C^{0,1}(A) \hookrightarrow C^0(\bar{A})$ is compact, i.e., if $(u_n)_{n \in \mathbb{N}}$ is a sequence of functions in $C^{0,1}(A)$, there exists a subsequence $(u_{n_k})_{k \in \mathbb{N}}$ of $(u_n)_{n \in \mathbb{N}}$ that converges uniformly in \bar{A} . \square

Proposition 2: Let $(u_n)_{n \in \mathbb{N}} \in C^{0,1}(A)$ be a sequence that converges uniformly to u and such that $[u_n]_1 < c$ for some $c > 0$. Then $u \in C^{0,1}(A)$ and $[u]_1 < c$. \square

Let us now consider (3), which is rewritten in the form of the more general Hamilton-Jacobi equation

$$\phi_t(x, t) + H(x, t, \nabla \phi(x, t)) = 0 \text{ in } \Omega \times (0, T) \quad (5)$$

where $H(x, t, p) = u(x, t)|p|$ is the Hamiltonian function. Consider also initial conditions $\phi(x, 0) = \phi_0(x)$ in $\bar{\Omega}$. In the following, we will focus on viscosity solutions “inside Ω ” and viscosity supersolutions in $\partial\Omega$ [35].

We need to assume the following.

Assumption 1: Let $u \in C^0(\bar{\Omega} \times [0, T])$ such that $u(x, t) > 0$ for $x \in N(\partial\Omega)$ and $t \in [0, T]$.¹ \square

Assumption 2: There exists $L > 0$ such that $|u(x, t) - u(y, t)| \leq L|x - y|$ for all $x, y \in \Omega$ and $t \in [0, T]$.² \square

Assumption 3: There exists $M > 0$ such that $|u(x, t_1) - u(x, t_2)| \leq M|t_1 - t_2|$ for all $x \in \bar{\Omega}$ and $t_1, t_2 \in [0, T]$.³ \square

Theorem 1: If $\phi_0 \in C^0(\bar{\Omega})$, there exists a viscosity solution $\phi \in C^0(\bar{\Omega} \times [0, T])$ for (5) such that $\phi(x, 0) = \phi_0(x)$ in $\bar{\Omega}$.

Proof. See [35, Theorem IV.2, p. 655] with all the required assumptions satisfied since more restrictive conditions hold owing to the specific choice of the Hamiltonian function. Moreover, such a result states that ϕ is a viscosity supersolution on $\bar{\Omega} \times (0, T)$. \square

It is worth noting that the viscosity solution $\phi \in C^0(\bar{\Omega} \times [0, T])$ is not unique in general. However, it is the minimum viscosity supersolution $v(x, t)$ of (5) on $\bar{\Omega} \times [0, T]$ such that $v(x, 0) \geq \phi_0(x)$ on $\bar{\Omega}$. Moreover, such a solution is Lipschitz w.r.t. x near $\partial\Omega$, as it follows from the proof of [35, Theorem IV.2, p. 655]. In fact, Assumption 1 holds and $H(x, t, p) \rightarrow +\infty$ as $|p| \rightarrow +\infty$ uniformly for $x \in N(\partial\Omega)$ and $t \in [0, T]$.

Let

$$\mathcal{F} := \{\phi \in C^0(\bar{\Omega} \times [0, T]) : \phi \text{ is Lipschitz on } N(\partial\Omega) \text{ and a solution of (5)}\}$$

and, for some $a > 0$,

$$\mathcal{U}_a := \{u \in C^{0,1}(\bar{\Omega} \times [0, T]) \text{ such that } u(x, t) \geq a, \\ x \in N(\partial\Omega), t \in [0, T]\}.$$

As it will be clearer from what follows, \mathcal{U}_a is the set of the admissible controls for a given cost functional that satisfies the next assumption.

Assumption 4: Let $J : \mathcal{U}_a \times \mathcal{F} \rightarrow [0, \infty)$ such that $J(\cdot, s) : \mathcal{U}_a \rightarrow [0, \infty)$ is lower semicontinuous for $s \in \mathbb{R}$.

Therefore, let us recast problem (4) as follows:

$$\inf_{u \in \mathcal{U}_a, \phi \in \mathcal{F}} J(u, \phi). \quad (6)$$

¹The positivity assumption in a neighborhood of $\partial\Omega$ allows one to rely on the existence of supersolutions in $\bar{\Omega}$ [35, Proposition II.2, p. 647].

²In principle, we may adopt a more general assumption, i.e., $|u(x, t) - u(y, t)| \leq \omega(|x - y|)$, where $\omega : [0, \infty) \rightarrow [0, \infty)$ is a continuous, nondecreasing, and subadditive function such that $\omega(0) = 0$ (see [35, Assumption (H2), p. 648]).

³Likewise in Assumption 2, we may relax such assumption (see [35, eq. (36), p. 655]).

Theorem 2: There exists $u^* \in \mathcal{U}_a$ such that

$$J(u^*, \phi^*) = \inf_{u \in \mathcal{U}_a, \phi \in \mathcal{F}} J(u, \phi)$$

for some $\phi^* \in \mathcal{F}$.

Proof. Since J is lower bounded, there exists a minimizing sequence $(u_k)_{k \in \mathbb{N}}$ for J in \mathcal{U}_a . For every $k \in \mathbb{N}$, let Φ_k be a viscosity solution of

$$\frac{\partial}{\partial t} \Phi_k(x, t) + u_k(x, t) |\nabla \Phi_k(x, t)| = 0 \text{ in } \Omega \times (0, T) \quad (7)$$

such that $\Phi_k(x, 0) = \phi_0(x)$, where $\Phi_k \in C^0(\overline{\Omega} \times [0, T])$ is Lipschitz on $N(\partial\Omega)$. Since u_k belongs to \mathcal{U}_a , from Propositions 1 and 2 it follows that there exists a subsequence of $(u_k)_{k \in \mathbb{N}}$ that uniformly converges in $\overline{\Omega} \times [0, T]$ to some $u^* \in C^{0,1}(\overline{\Omega} \times [0, T])$. To reduce the notational overhead and with a little abuse of notation, we will denote such a subsequence by u_k , and so it will be also for other sequences.

First of all, let us verify that u^* belongs to \mathcal{U}_a . Toward this end, notice that $u^*(x, t) \geq a$ for $x \in N(\partial\Omega)$ and $t \in [0, T]$. Since for every $\varepsilon > 0$ there exists $k_\varepsilon \in \mathbb{N}$ such that, for $k > k_\varepsilon$, it follows that $u^*(x, t) > u_k(x, t) - \varepsilon$ for $(x, t) \in \overline{\Omega} \times [0, T]$, if $x \in N(\partial\Omega)$ we obtain $u^*(x, t) > a - \varepsilon$. From the arbitrariness of ε it follows $u^*(x, t) \geq a$ for $x \in N(\partial\Omega)$ and $t \in [0, T]$ (from now on, we omit to recall the dependence on t for the sake of brevity).

Then, let us focus on (7), where u_k is the subsequence converging to u^* we considered before. Let us show that $\Phi_k \in C^0(\overline{\Omega} \times [0, T])$ converges to some $\phi^* \in C^0(\overline{\Omega} \times [0, T])$ such that

$$\frac{\partial}{\partial t} \phi^*(x, t) + u^*(x, t) |\nabla \phi^*(x, t)| = 0 \text{ in } \Omega \times (0, T) \quad (8)$$

with $\phi^*(x, 0) = \phi_0(x)$ by passing to a subsequence if necessary. Owing to the structure of the Hamiltonian $H(x, t, p)$ in (5) (it is convex in p and such that $H(x, t, p)$ tends to $+\infty$ as $|p| \rightarrow +\infty$ uniformly for $x \in N(\partial\Omega)$), there exist $\alpha > 0$ and $\beta > 0$ such that $H(x, t, p) \geq \alpha|p| - \beta$. From such inequality and Assumption 3, it follows that $\nabla \Phi_k$ is uniformly bounded w.r.t. k in some $N(\partial\Omega)$. Thus, using (8) we get that also Φ_k is uniformly bounded, and hence there exists $\delta > 0$ such that $\|\Phi_k\|_1 \leq \delta$ in some $N(\partial\Omega)$. Using the arguments in [36, Theorem 1, p. 385], it follows that an a-priori estimate of the modulus of continuity of Φ_k near $\partial\Omega$ propagates in Ω . Thus, Φ_k is bounded in $C^{0,1}(\overline{\Omega} \times [0, T])$ and, thanks to Propositions 1 and 2, it admits a subsequence that uniformly converges to some ϕ^* , which is Lipschitz in some $N(\partial\Omega)$. By passing to a further subsequence if necessary, for u_k and Φ_k , using [37, Theorem 1.4, p. 375], we obtain (8).

If Φ_k is a viscosity supersolution in $\overline{\Omega} \times (0, T)$ and it is Lipschitz in $N(\partial\Omega)$, then u_k is Lipschitz in $\overline{\Omega}$, Φ_k and u_k admit subsequences that uniformly converge to ϕ^* and u^* , ϕ^* is a viscosity supersolution in $\overline{\Omega} \times (0, T)$ of (8). Therefore, we finally obtain that ϕ^* belongs to \mathcal{F} and, using [38, Proposition 7.1.2, p. 206] with Assumption 4, we get

$$J(u^*, \phi^*) \leq \liminf_{k \rightarrow +\infty} J(u_k, \Phi_k)$$

which concludes the proof. \square

Remark 1: It is worth noting that, since $\overline{\Omega} \times [0, T]$ is compact and $u^* \in C^{0,1}(\overline{\Omega} \times [0, T])$, it follows that u^* belongs to $L^\infty(\overline{\Omega} \times [0, T])$ and, owing to the fact that the measure of $\overline{\Omega} \times [0, T]$ is finite, we obtain $u^* \in L^p(\overline{\Omega} \times [0, T])$ for every $p \geq 1$.

Unfortunately, in general it is difficult to find an analytic expression for the solution u^* . This motivates the use of methods to search for approximate solutions, as detailed in Section III.

III. SEARCH FOR APPROXIMATE SOLUTIONS

The optimal control problem of LSs (6) is of functional optimization since the unknown is a function, i.e., the optimal control law that drives the propagating front. For this kind of problems, the idea of finding approximate solutions with the ERIM has been applied in the past [17]–[20]. It consists in searching for approximations of the optimal solution by forcing the unknown control action to take on a parameterized structure, and then tuning its parameters to minimize a given index cost. In our case, the goal consists in approximating the unknown mapping $(x, t) \mapsto u^*(x, t)$ that is the solution of the optimization problem (6). Toward this end, we consider linear combinations of parameterized basis functions as fixed structures for the control law as in (9). It is known that such structures guarantee a good compromise between approximation accuracy and computational effort required for the tuning of the parameters (see, e.g., [39], [40]):

$$\gamma(\cdot, w) = \sum_{i=1}^n c_i \psi(\cdot, \kappa_i) + b, \quad c_i \in \mathbb{R}, \quad b \in \mathbb{R}, \quad \kappa_i \in \mathbb{R}^l \quad (9)$$

where ψ is a parameterized basis function and the parameters (or weights) to be optimized are the components of the vector $w := (c, b, \kappa) \in \mathbb{R}^{N(n)}$, where $c := (c_1, c_2, \dots, c_n)$, $\kappa := (\kappa_1, \kappa_2, \dots, \kappa_l)$, and $N(n) = n(1 + l) + 1$. Most of approximating functions commonly used in the literature belongs to the class (9), such as feedforward neural networks, radial-basis-functions with adjustable centers and widths, free-node splines, and trigonometric polynomials with free frequencies and phases.

Generally speaking, the set of approximating functions in (9) is required to be dense in the space of functions where we search for the solution to our problem, for which a solution in L^p is proved to exist, as pointed out in Remark 1. Another important feature is the so-called “universal approximation property,” which means that the unknown continuous mapping can be approximated arbitrarily well for some choice of the vector of parameters w [41]–[43]. Such a property is satisfied by a large family of approximating functions, including the above-introduced ones.

According to the ERIM paradigm, in order to solve (6) we have to force

$$u(x, t) = \gamma(x, t, w) \quad (10)$$

in the normal flow equation (3) and cost functional that, from now on, we choose as follows:

$$J(u, \phi) = \int_0^T \int_{\Omega} h(\phi(x, t), u(x, t), t) dx dt + \int_{\Omega} \bar{h}(\phi(x, T)) dx \quad (11)$$

where $h : \mathbb{R} \times \mathbb{R} \times [0, T] \rightarrow \mathbb{R}$ and $\bar{h} : \mathbb{R} \rightarrow \mathbb{R}$ is a final penalty term. The evolution over time and space of a LS of ϕ is shaped by using the performance index (11), which in general may depend on the interior or the boundary of the interface (see, e.g., [16]).

As a consequence of (10), both ϕ and u depends on the choice of w . Hence, also J turns out to be a function of w . From now on, according to the context and with a little abuse of notation, we will highlight the dependence of J on either ϕ and u or simply w . Thus, the original functional optimization problem (6) is converted into a mathematical programming one that consists in the search for the optimal weights w^o minimizing the cost J , i.e.,

$$w^o \in \arg \min_{w \in \mathbb{R}^{N(n)}} J(w). \quad (12)$$

In Section IV, we propose two techniques to find a solution of problem (12). Both methods require to compute the gradient of J w.r.t. the parameters w of the approximating function γ in (9). Hence, in the following we compute the exact, analytic expression for this gradient. Toward this end, we need to assume the following.

Assumption 5: The functions $h : \mathbb{R} \times \mathbb{R} \times [0, T] \rightarrow \mathbb{R}$ and $\bar{h} : \mathbb{R} \rightarrow \mathbb{R}$ are continuously differentiable. \square

Based on the aforesaid, we can state the following.

Proposition 3: The gradient w.r.t. the weights of the cost functional is

$$\nabla_w J(u, \phi) = \int_0^T \int_{\Omega} \left(h_u(\phi, \gamma) + \mu |\nabla \phi| \right) \nabla_w \gamma dx dt \quad (13)$$

where the mapping $\mu : \bar{\Omega} \times [0, T] \rightarrow \mathbb{R}$ satisfies the adjoint equation

$$-\mu_t = (\mu \gamma F_1)_x + (\mu \gamma F_2)_y - h_\phi(\phi, \gamma) \text{ in } \Omega \times (0, T) \quad (14)$$

where $\mu(x, T) = -\bar{h}_\phi(\phi(x, T))$ in $\bar{\Omega}$.

Proof. To reduce the notational burden, from now on we will drop the dependence on x and t and write explicitly the dependence on w . Let $\tilde{\phi}(w, \tilde{w}) := \phi(w + \tilde{w}) - \phi(w) \in \mathcal{F}$, where $\tilde{w} \in \mathbb{R}^{N(n)}$. Of course, if $\tilde{w} \rightarrow 0$ also $\tilde{\phi}$ tends to zero.

Using the Fréchet derivative of the cost along the direction $(\tilde{\phi}, \tilde{w})$, from (3) it follows that

$$\phi_t(w + \tilde{w}) + \gamma(w + \tilde{w}) |\nabla \phi(w + \tilde{w})| = 0 \quad (15a)$$

$$\phi_t(w) + \gamma(w) |\nabla \phi(w)| = 0. \quad (15b)$$

After replacing $\gamma(w + \tilde{w})$ with a Taylor expansion of the first order centered in w and using the same approximation for the norm of the gradient of ϕ , i.e.,

$$|\nabla \phi(w + \tilde{w})| = |\nabla \phi(w)| + \frac{\nabla \phi(w)}{|\nabla \phi(w)|} (\tilde{\phi}_x, \tilde{\phi}_y) + r_0$$

where $\tilde{w} \mapsto r_0(w, \tilde{w})$ is a remainder of order higher than one, from (15) it follows that

$$\tilde{\phi}_t + |\nabla \phi| \nabla_w \gamma \tilde{w} + \gamma F \cdot \tilde{\Phi} + r_1 = 0 \quad (16)$$

where, adopting the same notation of [29], we let $F := (F_1, F_2)$ with $F_1 := \phi_x / |\nabla \phi|$, $F_2 := \phi_y / |\nabla \phi|$, $\tilde{\Phi} := (\tilde{\phi}_x, \tilde{\phi}_y)$, and $\tilde{w} \mapsto r_1(w, \tilde{w})$ accounts for all the remainders of order higher than one.

In order to compute the derivative of J in (w, ϕ) along the direction $(\tilde{w}, \tilde{\phi})$, we apply a Taylor expansion of the terms inside the integrals, i.e., we get

$$J(w + \tilde{w}, \phi + \tilde{\phi}) - J(w, \phi) = \int_0^T \int_{\Omega} h_\phi(\phi, \gamma) \tilde{\phi} dx dt + \int_0^T \int_{\Omega} h_u(\phi, \gamma) \nabla_w \gamma \tilde{w} dx dt + \int_{\Omega} \bar{h}_\phi(\phi) \tilde{\phi} dx + r_2 \quad (17)$$

where we have highlighted the dependence of u on w , and $\tilde{w} \mapsto r_2(w, \tilde{w})$ is a remainder of order higher than one. The goal is to find the first-order necessary condition of optimality by using the first variation with (16) as a constraint. First of all, we introduce $(x, t) \mapsto \mu(x, t)$ as Lagrange multiplier. Then, we add the product between $\mu(x, t)$ and (16) to the right hand side of (17). If we integrate on $\Omega \times [0, T]$, we can write

$$J(w + \tilde{w}, \phi + \tilde{\phi}) - J(w, \phi) = \int_0^T \int_{\Omega} h_\phi(\phi, \gamma) \tilde{\phi} dx dt + \int_0^T \int_{\Omega} h_u(\phi, \gamma) \nabla_w \gamma \tilde{w} dx dt + \int_{\Omega} \bar{h}_\phi(\phi) \tilde{\phi} dx + \int_0^T \int_{\Omega} \mu \left(\tilde{\phi}_t + |\nabla \phi| \nabla_w \gamma \tilde{w} + \gamma F \cdot \tilde{\Phi} \right) dx dt + r_3 \quad (18)$$

where the remainder r_3 accounts for both r_1 and r_2 . From

$$\mu \tilde{\phi}_t = \left(\mu \tilde{\phi} \right)_t - \mu_t \tilde{\phi}$$

it follows that

$$\begin{aligned} \int_0^T \int_{\Omega} \mu \tilde{\phi}_t dx dt &= \int_0^T \int_{\Omega} \left(\mu \tilde{\phi} \right)_t - \mu_t \tilde{\phi} dx dt \\ &= \int_{\Omega} \mu(x, T) \tilde{\phi}(x, T) - \mu(x, 0) \tilde{\phi}(x, 0) dx \\ &\quad - \int_0^T \int_{\Omega} \mu_t \tilde{\phi} dx dt \end{aligned} \quad (19)$$

after a change of the order of integration in the first term. Using the Green identity and imposing a null $\tilde{\phi}$ on the boundary of Ω , we have

$$\begin{aligned} \int_0^T \int_{\Omega} \mu \gamma F \cdot \tilde{\Phi} dx dt &= \int_0^T \int_{\Omega} (\mu \gamma F_1) \tilde{\phi}_x \\ &\quad + (\mu \gamma F_2) \tilde{\phi}_y dx dt = \int_0^T \int_{\partial \Omega} (\mu \gamma F_1 + \mu \gamma F_2) \tilde{\phi} dx dt \\ &\quad - \int_0^T \int_{\Omega} (\mu \gamma F_1)_x \tilde{\phi} + (\mu \gamma F_2)_y \tilde{\phi} dx dt \\ &= - \int_0^T \int_{\Omega} (\mu \gamma F_1)_x \tilde{\phi} + (\mu \gamma F_2)_y \tilde{\phi} dx dt. \end{aligned} \quad (20)$$

After replacing (19) and (20) in (18), we obtain

$$\begin{aligned}
J(w + \tilde{w}, \phi + \tilde{\phi}) - J(w, \phi) &= \int_0^T \int_{\Omega} (h_{\phi}(\phi, \gamma) \\
&\quad - \mu_t - (\mu \gamma F_1)_x - (\mu \gamma F_2)_y) \tilde{\phi} dx dt \\
&\quad + \int_0^T \int_{\Omega} (h_u(\phi, \gamma) \nabla_w \gamma + \mu |\nabla \phi| \nabla_w \gamma) \tilde{w} dx dt \\
&\quad + \int_{\Omega} (\bar{h}_{\phi}(\phi(x, T)) \tilde{\phi}(x, T) + \mu(x, T) \tilde{\phi}(x, T) \\
&\quad - \mu(x, 0) \tilde{\phi}(x, 0)) dx + r_3.
\end{aligned}$$

If we choose $\tilde{\phi}(x, 0) = 0$ on Ω and solve the adjoint equation (14), then the gradient w.r.t. the weights of the cost J is given by (13). \square

The solution of the optimization problem (12) requires the use of efficient numerical techniques to solve both the forward normal flow equation (3) and the backward adjoint equation (14). It is worth noting that the numerical schemes for the two equations cannot be the same, as the structure of the adjoint equation is very different from the forward one. It is known in the literature that the use of adjoint equations for the control of PDEs drastically reduces the computational time [9], [29]. The price to pay is a considerable increase in the effort to correctly discretize the adjoint equation, also in term of storage capacity, as the solution of the forward equation has to be saved for a large number of time steps. In our case, the normal flow equation (3) and the adjoint equation (14) are both of hyperbolic type. As a consequence, we have to use high-order finite-difference schemes for hyperbolic PDEs, as it will be detailed in Section V. Moreover, imposing the correct boundary conditions in hyperbolic equations is a nontrivial task in general. This is true also for the adjoint equation (14), as the source term $h_{\phi}(\phi, \gamma(w))$ may have a large variability for values of w far from the global optimum, which can create spurious and non-physical reflections at the boundary.

IV. SELECTION OF THE OPTIMAL WEIGHTS

In this section, we present two optimization methods to find a solution to problem (12), i.e., train the approximating structure γ in (9). Both methods exploit the computation of the gradient of J using (13) and (14). The first one is based on line-search descent methods, whereas the second one relies on an approach based on the Newton method [44]. Simulation results will be shown in Section V to evaluate the robustness of such methods w.r.t. local minima trapping.

A. Line-search Descent Optimization

Generally speaking, descent methods are given by recursive algorithms having the following structure:

$$w_{k+1} = w_k + \alpha_k d_k, \quad k = 0, 1, \dots$$

where $\alpha_k > 0$ is the descent step and $d_k \in \mathbb{R}^{N(n)}$ is the descent direction. Specifically, they are referred to as line-search descent methods if, at each iteration k , the step α_k is chosen via a line search. The descent direction d_k is usually selected by using the information on the gradient of the

cost to be minimized. The specific choices depend on the particular descent method adopted. Without loss of generality, let us consider the classic steepest-descent algorithm [44]. In this case, the descent direction is equal to the opposite of the gradient (13), i.e., $d_k = -\nabla_w J(w_k)$. The step α_k is chosen through a line search that consists in minimizing $J(w_k - \alpha_k \nabla_w J(w_k))$ w.r.t. α_k .

B. Newton-based Optimization

The standard Newton method does not require line search since it consists in iterating

$$w_{k+1} = w_k + (\nabla^2 J(w_k))^{-1} \nabla J(w_k), \quad k = 0, 1, \dots$$

where $\nabla^2 J(w_k)$ is the Hessian of $J(w_k)$. To reduce the computational effort, the idea behind quasi-Newton methods is that of avoiding the computation of the Hessian by using some local approximation around the point of optimum [44]. Toward this end, the Gauss-Newton method is a quasi-Newton approach for least squares problems that allows one to treat a large amount of data in an efficient way [10]. Under linear assumptions, the least squares problem can be solved recursively by an iterative algorithm that coincides with a Kalman filter [45]. In the nonlinear case, one can resort to the EKF, and hence regard the optimization as an EKF learning task [31]. It is worth noting that the convergence properties of the EKF are still quite unknown at present. Only results on the boundedness of the expected quadratic error are reported in the literature [46]. In spite of its poor theoretical foundation, a number of successful results concerning the application of the EKF to neural network training are available [24]–[28]. Finally, it is worth noting that, in our context and likewise for line-search descent methods, the Newton and quasi-Newton methods are not ensured to converge to a global optimum since the optimization problem we have to solve is not convex. Thus, there exists the need of dealing with local minima. From now on, we will refer to Gauss-Newton-based techniques as EKF optimization methods according to the parlance of the community working in the area of neural networks and learning systems.

C. Evaluation of Robustness w.r.t. Local Minima Trapping

As said, the performances of both line-search descent and EKF-based methods may be undermined by local minima that prevent from converging to a global optimum. To evaluate how such techniques are able to avoid local minima, we adopt a multistart procedure that consists in randomly choosing L different initial weights and apply the same optimization method for each value of the initial weights in order to obtain a performance index as low as possible. Such a technique is shown in Procedure 1. The stopping criteria consist in finding a “small” norm of either the gradient of the cost or the difference between the estimated parameters in two consecutive iterations, together with a maximum number of iterations. Let $J^{(l)*}$ and $w^{(l)*}$ denote the optimal cost and parameters corresponding to the l -th initial guess, respectively. Furthermore, let $\bar{T}^{(l)}$ be the time needed to find the optimal weights starting from

Procedure 1 (Multistart Simulation Optimization)

```

1: for  $l$  from 1 to  $L$  do
2:   generate a random initial choice of  $w_0^{(l)}$ 
3:    $k \leftarrow 0$ 
4:   Reset execution time
5:   while (stopping criteria are not satisfied) do
6:      $u(x, t) \leftarrow \gamma(x, t, w_k^{(l)})$ 
7:     solve the normal flow equation (3)
8:     solve the adjoint equation (14)
9:     compute the gradient (13)
10:     $w_{k+1}^{(l)} \leftarrow$  line-search descent (or EKF-based)
        step starting from  $w_k^{(l)}$ 
11:     $k \leftarrow k + 1$ 
12:  end while
13:   $\bar{T}^{(l)} \leftarrow$  execution time of the while loop
14:   $w^{(l)*} \leftarrow w_k^{(l)}$ 
15:   $J^{(l)*} \leftarrow J(w_k^{(l)})$ 
16: end for
17:  $J^* \leftarrow \min_{l=1, \dots, L} (J^{(l)*})$ 
18:  $w^* \leftarrow \arg \min_{l=1, \dots, L} (J^{(l)*})$ 

```

the l -th initial choice. Then, the parameters corresponding to the lower value of the cost are selected as the optimal ones. In other words, $w^* := \arg \min_{l=1, \dots, L} (J^{(l)*})$ denotes the best parameter vector and $J^* := \min_{l=1, \dots, L} (J^{(l)*})$ is the corresponding optimal cost.

V. SIMULATION RESULTS

In this section, we present the simulation results obtained in two different numerical examples concerning the tracking of a reference curve in both two and three dimensions.

In particular, our aim is to determine a control function $u(x, t)$ that drives the zero LS $\Gamma_0(t)$ of the solution $\phi(x, t)$ of the normal flow equation (3) in order to match the zero LS $\Gamma_0^{\text{ref}}(t)$ of a given reference function $\phi_{\text{ref}}(x, t)$. The reference $\Gamma_0^{\text{ref}}(t)$ can be interpreted as a desired shape in Ω that changes over time depending on $\phi_{\text{ref}}(x, t)$. Without loss of generality, we focused on a tracking performance index computing the difference between the reference and approximate LSs as follows:

$$J = \int_0^T \eta (\Gamma_0(t) \Delta \Gamma_0^{\text{ref}}(t))^2 dt \quad (21)$$

where Δ is the symmetric difference operator, i.e., $A \Delta B = (A \cup B) \setminus (A \cap B)$, and η corresponds to an outer measure on \mathbb{R}^q . Figure 2 shows a sketch of the area to minimize at a generic time instant in two dimensions. However, notice that the proposed approach is valid also for other performance indexes of the same kind of (11) that do not require $\Gamma_0^{\text{ref}}(t)$ or $\phi_{\text{ref}}(x, t)$.

Since in general computing the symmetric difference is a difficult task [3], we minimize the following cost instead of (21):

$$J = \int_0^T \int_{\Omega} \left(\hat{H}(\phi(x, t)) - \hat{H}(\phi_{\text{ref}}(x, t)) \right)^2 dx dt \quad (22)$$

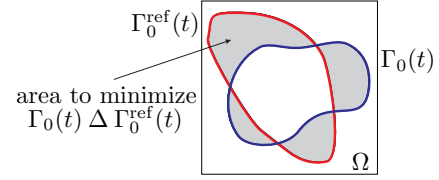


Fig. 2. Front tracking problem at a generic time instant t in two dimensions.

where $\hat{H}(\cdot)$ is an approximation of the Heaviside step function given by

$$\hat{H}(\xi) := \frac{1}{2} + \frac{1}{2} \tanh\left(\frac{\xi}{\tau}\right).$$

The coefficient τ tunes the smoothness of the approximation. Specifically, we fixed τ equal to 10^{-2} . Notice that (22) is a particular version of (11) with null final cost.

As regards the 2D example, for the numerical solution of (3) we chose a space domain $\Omega = (-0.75, +0.75) \times (-0.5, +0.5)$ discretized by using a regular grid of 75×50 points and a time interval equal to $[0, 1.5]$ with sampling time $\Delta t = 0.03$. Concerning the 3D case, we adopted a domain $\Omega = (-1.5, 1.5) \times (-1.0, 1.0) \times (-1.0, 1.0)$ sampled with a grid made of $90 \times 60 \times 60$ nodes. The time interval of the simulation was $[0, 1.5]$ sampled with a time step $\Delta t = 0.05$, i.e., we needed 30 time steps to perform the simulation. We chose one-hidden-layer feedforward neural networks with sigmoidal activation functions as parameterized structures γ in (9). In particular, we tested various numbers of basis functions, i.e., we considered $n = 5, 10$, and 15 neurons to ensure a sufficient accuracy with quite simple approximating structures. Both the line-search and EKF-based optimization methods were compared by applying Procedure 1 with $L = 50$ different initial weights. The tolerance for the stopping criteria and the maximum number of iterations were chosen equal to 10^{-9} and 1000 , respectively.

The simulations were carried out on a personal computer with a 2.6 GHz Intel Xeon CPU with 64 GB of RAM. The line-search minimizations were executed through the *fmincon* function contained in the Matlab Optimization Toolbox, which implements both the interior-point and sequential quadratic programming algorithms. Concerning the EKF-based algorithm, an experimental tuning was performed to select the covariance matrices in order to obtain the fastest convergence to the optimal cost [31].

The normal flow equation (3) and the corresponding adjoint equation (14) for the computation of the gradient of the cost were solved numerically by using the Matlab toolbox implemented by Mitchell [47], which includes various solvers of Hamilton-Jacobi equations. More specifically, for the discretization of the normal flow equation we employed an upwind second-order essentially non-oscillatory scheme [6, chap. 3] w.r.t. space. Concerning the time approximation, we used a total variation diminishing Runge-Kutta scheme of second order. Notice that the convective flux terms in the adjoint equation (14) depend explicitly on x . Unfortunately, Mitchell's toolbox does not take into account such a dependency, and

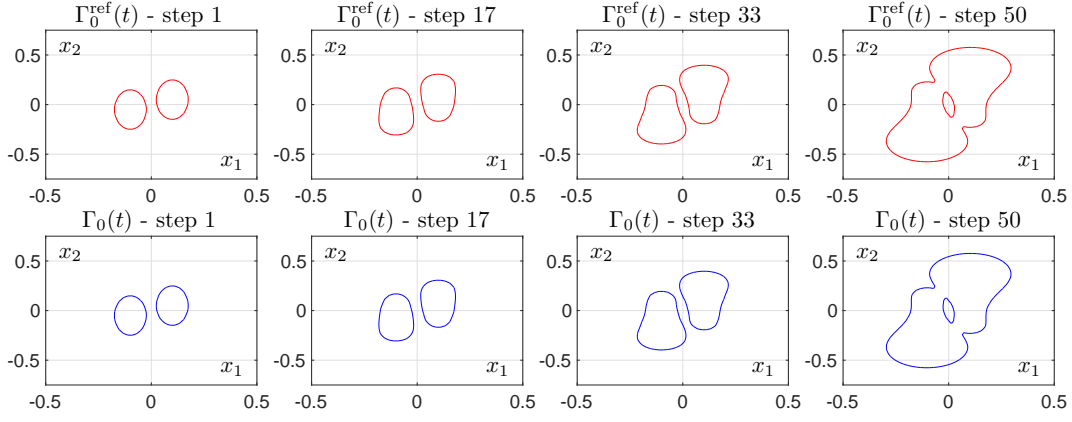


Fig. 3. Front tracking snapshots obtained with the EKF-based optimization in the 2D example with $n = 10$ basis functions.

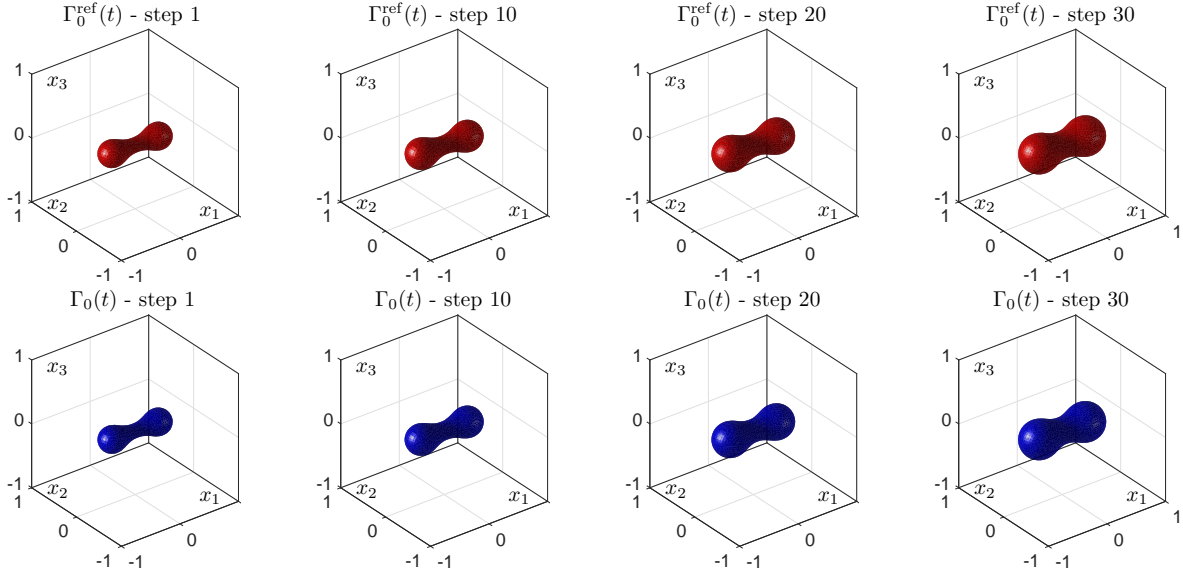


Fig. 4. Front tracking snapshots obtained with the EKF-based optimization in the 3D example with $n = 10$ basis functions.

therefore we modified the numerical scheme by deriving the flux and adding a source term.

Figures 3 and 4 show the results of the tracking in the 2D and 3D cases, respectively, using the EKF-based optimization algorithm. The number n of basis functions is equal to 10 in all the cases. Similar results could be displayed for the line-search optimization method and for the other considered numbers of neurons, but they are not reported here for the sake of compactness. In particular, the reference and simulated fronts (red and blue plots, respectively) at certain time steps are shown. In both the 2D and 3D cases, the reference and simulated fronts match with a great accuracy. Notice that a change of topology occurs in the 2D example, as two ellipses join into a unique curve.

Table I shows a summary of the simulation results. More specifically, it showcases the means of the optimal costs $J^{(l)*}$ and of the times $\bar{T}^{(l)}$, $l = 1, \dots, L$, obtained by applying Procedure 1 for the line-search descent and EKF-based optimizations. Figure 5 displays the boxplots of the costs $J^{(l)*}$ and of the times $\bar{T}^{(l)}$ obtained with 10 neurons.

TABLE I
SUMMARY OF THE SIMULATION RESULTS.

| | n | mean of $J^{(l)*}$ | | mean of $\bar{T}^{(l)}$ (seconds) | |
|----|-----|----------------------|----------------------|-----------------------------------|-------------------|
| | | line-search opt | EKF opt | line-search opt | EKF opt |
| 2D | 5 | $2.46 \cdot 10^{-3}$ | $1.00 \cdot 10^{-4}$ | $3.20 \cdot 10^2$ | $1.00 \cdot 10^3$ |
| | 10 | $3.30 \cdot 10^{-3}$ | $4.66 \cdot 10^{-6}$ | $5.82 \cdot 10^2$ | $1.14 \cdot 10^3$ |
| | 15 | $1.07 \cdot 10^{-3}$ | $1.47 \cdot 10^{-6}$ | $9.28 \cdot 10^2$ | $1.53 \cdot 10^3$ |
| 3D | 5 | $8.86 \cdot 10^{-4}$ | $5.23 \cdot 10^{-4}$ | $4.40 \cdot 10^3$ | $3.49 \cdot 10^4$ |
| | 10 | $8.63 \cdot 10^{-4}$ | $1.85 \cdot 10^{-4}$ | $4.10 \cdot 10^3$ | $3.42 \cdot 10^4$ |
| | 15 | $8.32 \cdot 10^{-4}$ | $1.81 \cdot 10^{-4}$ | $3.96 \cdot 10^3$ | $3.43 \cdot 10^4$ |

The simulations confirm that the two proposed approaches to select the optimal weights are both valid, as the minimum values of the costs obtained starting from the considered 50 initial guesses are similar. The most important difference is that the line-search optimization presents a larger dispersion around the medians compared to the EKF-based algorithm. The reason of this behavior is due to the fact that the former optimization may end up in local minima more frequently depending on the initial guess. On the contrary, the EKF

optimization does not suffer from such an issue, as it provides almost the same values of the cost starting from all the different initial weights. Thus, the results suggest that starting from many different initial parameters is actually useless for the EKF optimization since always almost the same optimal costs are obtained.

In general, the times $\bar{T}^{(l)}$ of the EKF are larger than those of the line-search optimization. This may be ascribed to the fact that the stopping criteria of the latter are easily satisfied if the minimization procedure is trapped into a local minimum. However, owing to the above-discussed robustness w.r.t. local minima trapping of the EKF approach, a reduction of the overall computational time needed to approximate the optimal control law can be achieved using this method. In fact, without loss of generality, consider the case of $n = 10$ neurons in the 2D example. The overall time required to find the optimal weights with the line-search optimization starting from 50 different initial guesses is equal to $5.82 \cdot 10^2 \times 50 \simeq 2.91 \cdot 10^4$ seconds on the average. Instead, the same time for the EKF starting from a single initial guess is equal to $1.14 \cdot 10^3$ seconds, with a saving of about the 95% of the computational time.

For both the considered optimization approaches, the optimal costs do not sensibly vary with the number n of basis functions, a part from the case of $n = 5$ for the EKF optimization in the 2D example, even if a slight reduction of the costs is experienced with an increase of the number of neurons. This suggests that all the considered values for n are enough to obtain satisfactory approximations.

Summarizing, the EKF-based algorithm appears to be more robust to avoid local minima trapping w.r.t. the line-search one in both the 2D and 3D cases, which makes it more well-suited to being used in the presence of large local variations of the cost.

A. Performances of the Controller Under Model Uncertainties

The performances of the approximate controllers obtained using either the line-search or the EKF optimization in the 2D and 3D examples were evaluated also in the presence of uncertainties in the normal flow equation (3). More specifically, we applied the control law obtained with both optimization methods to the following equation instead of (3):

$$\phi_t(x, t) + u |\nabla \phi(x, t)| = \xi(x, t) \quad (23)$$

where $\xi : \bar{\Omega} \times [0, T] \rightarrow \mathbb{R}$ is a disturbance acting as the source term of the equation. Without loss of generality, we assumed that this noise has a Gaussian probability distribution with zero mean and fixed variance for all $t \in [0, T]$. In particular, we evaluated the effect of the uncertainty for increasing values of the variance, i.e., we varied it from 10^{-5} up to 10^{-1} . To give statistical significance to the results, we considered all the $L = 50$ vectors of weights obtained in correspondence to the different initial guesses.

Figure 6 contains the boxplots of the optimal costs in the case of $n = 10$ basis functions. For the sake of comparison, in the figure the boxplots of the cost obtained in the absence of uncertainty are also reported, and denoted by $\text{var}(\xi) = 0$. It turns out that the approximate controllers obtained in

the noise-free case guarantee good performances also in the presence of uncertainties in the source term of the equation up to values of the variance equal to 10^{-2} . In fact, the optimal costs are near to those obtained in the absence of noises. For greater variances, a large increase of the costs can be observed. The superiority of the EKF optimization approach w.r.t. the line-search one is preserved also in the presence of disturbances. In fact, notice that in all the cases the boxplots of the former method are characterized by a lower median and a reduced dispersion around the median w.r.t. those of the latter.

This is quite a satisfactory result, as it indicates that it is possible to train the approximating networks in the noise-free case and then use the optimal weights also in the presence of uncertainties without significant decays of performances, provided the uncertainty is not too large.

VI. CONCLUSIONS

In this paper, we have investigated an approach based on the ERIM to find approximate solutions to the optimal control problem of propagating fronts associated with the level sets of the normal flow equation. The optimization of the cost functional related to the performance of the control policy has been performed subject to the dynamics of such equation as a constraint. Two different techniques have been presented to select the parameters of the approximate control policy, both exploiting the gradient of the cost w.r.t. the weights of the control action, computed by solving the related adjoint equation backwards in time. The first algorithm is based on line-search methods, whereas the second technique is a quasi-Newton method that can be regarded as an EKF learning in the wide research field of neural networks. The EKF-based optimization has turned out to be more robust w.r.t. the trapping into local minima, as shown via numerical simulations in two and three dimensional examples.

Future works will be devoted to apply the proposed approaches to control the LS dynamics over an infinite horizon by studying the stability of the resulting controllers. Lastly, we will investigate the application of the techniques presented in this paper to real-world systems involving the control of moving fronts with a cascade of PDEs, describing the physical phenomenon together with the LS one.

ACKNOWLEDGMENT

The authors would like to thank Dr. Anna Rossi for helpful comments and suggestions.

REFERENCES

- [1] S. Osher and J. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations," *J. Comput. Phys.*, vol. 79, no. 1, pp. 12–49, 1988.
- [2] J. Sethian, *Level Set Methods and Fast Marching Methods*, 2nd ed. Cambridge: Cambridge University Press, 1999, vol. 3.
- [3] J. Berg and K. Holmström, "On parameter estimation using level sets," *SIAM J. Control Optim.*, vol. 37, no. 5, pp. 1372–1393, 1999.
- [4] J. Sethian and P. Smereka, "Level set methods for fluid interfaces," *Annual Rev. Fluid Mechanics*, vol. 35, no. 1, pp. 341–372, 2003.
- [5] S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*, ser. Applied Mathematical Sciences. New York: Springer-Verlag, 2003, vol. 153.

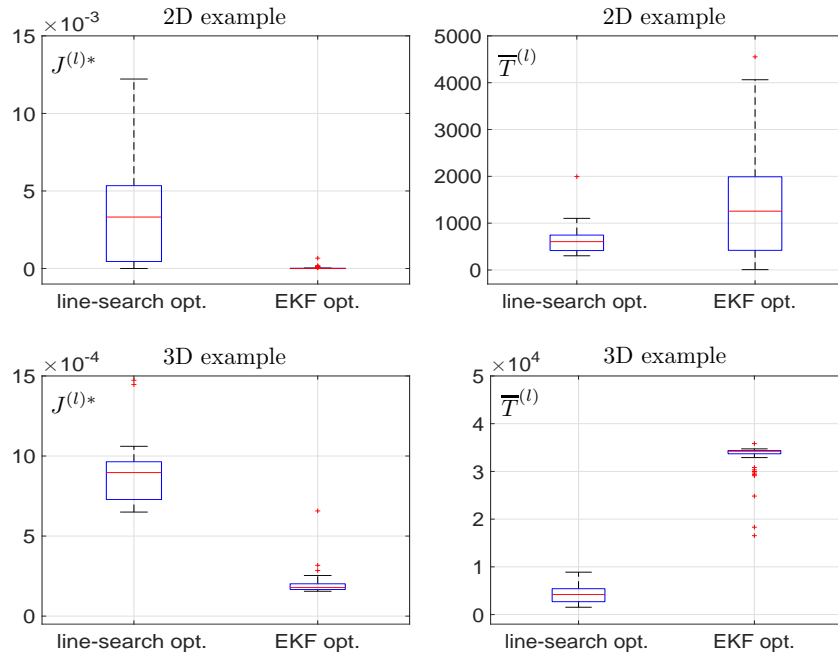


Fig. 5. Boxplots of the optimal cost and of the optimization time obtained with $n = 10$ basis functions.

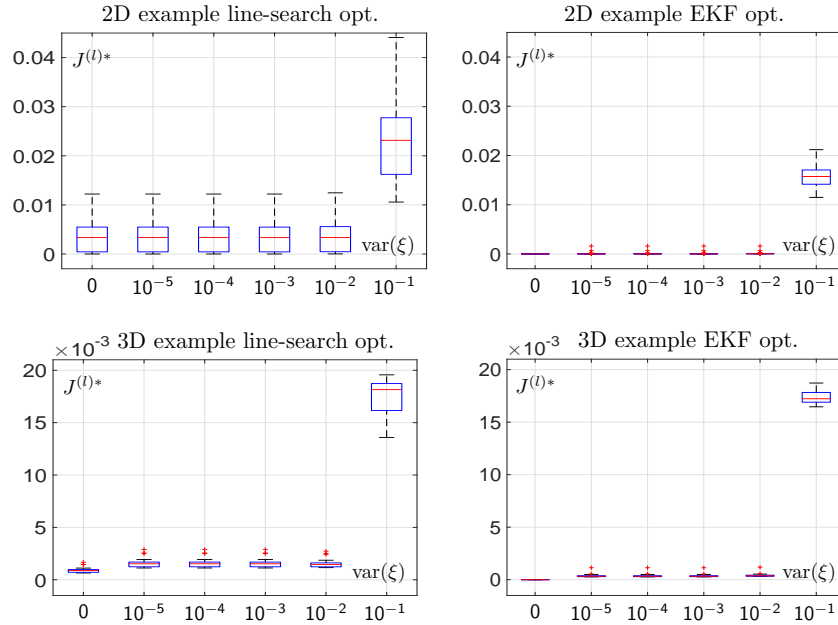


Fig. 6. Boxplots of the optimal costs obtained with $n = 10$ basis functions in the presence of increasing uncertainties in the model.

- [6] R. Kimmel, *Numerical Geometry of Images*. New York: Springer-Verlag, 2004.
- [7] C. Hirt and B. Nichols, "Volume of fluid (VOF) method for the dynamics of free boundaries," *J. Comput. Phys.*, vol. 39, no. 1, pp. 201–225, 1981.
- [8] A. Alessandri, P. Bagnerini, and M. Gaggero, "Optimal control of level sets dynamics," in *Proc. American Control Conf.*, 2014, pp. 2208–2213.
- [9] A. Alessandri, P. Bagnerini, M. Gaggero, and A. Traverso, "Further results on the optimal control of fronts generated by level set methods," in *Proc. American Control Conf.*, 2016, pp. 5225–5230.
- [10] D. Bertsekas, "Incremental least-squares methods and the extended Kalman filter," *SIAM J. on Optimization*, vol. 6, no. 3, pp. 807–822, 1996.
- [11] L. Bronsard and B. Wetton, "A numerical method for tracking curve networks moving with curvature motion," *J. Comput. Physics*, vol. 120, no. 1, pp. 66–87, 1995.
- [12] Y. Kim, M.-C. Lai, and C. Peskin, "Numerical simulations of two-dimensional foam by the immersed boundary method," *J. Comput. Physics*, vol. 229, no. 13, pp. 5194–5207, 2010.
- [13] J. Sethian and A. Wiegmann, "Structural boundary design via level set and immersed interface methods," *J. Comput. Phys.*, vol. 163, no. 2, pp. 489–528, 2000.
- [14] M. Haque, A. Rahmani, M. Egerstedt, and A. Yezzi, "Efficient foraging strategies in multi-agent systems through curve evolutions," *IEEE Trans. Autom. Control*, vol. 59, no. 4, pp. 1036–1041, 2014.
- [15] M. Hinze and S. Ziegenbalg, "Optimal control of the free boundary in a two-phase Stefan problem," *J. Comput. Phys.*, vol. 223, no. 2, pp. 657–684, 2007.
- [16] M. Bernauer and R. Herzog, "Optimal control of the classical two-phase Stefan problem in level set formulation," *SIAM J. Sci. Comput.*, vol. 33, no. 1, pp. 342–363, 2011.

- [17] R. Zoppoli and T. Parisini, "Learning techniques and neural networks for the solution of N-stage nonlinear nonquadratic optimal control problems," in *Systems, Models and Feedback: Theory and Applications*, A. Isidori and T. J. Tarn, Eds. Birkhäuser, Boston, 1992, pp. 193–210.
- [18] T. Parisini and R. Zoppoli, "Neural networks for feedback feedforward nonlinear control systems," *IEEE Trans. Neural Netw.*, vol. 5, no. 3, pp. 436–449, 1994.
- [19] T. Parisini and R. Zoppoli, "Neural approximations for multistage optimal control of nonlinear stochastic systems," *IEEE Trans. Automatic Control*, vol. 41, no. 6, pp. 889–895, 1996.
- [20] R. Zoppoli, M. Sanguineti, and T. Parisini, "Approximating networks and extended Ritz method for the solution of functional optimization problems," *J. Optim. Theory Appl.*, vol. 112, no. 2, pp. 403–440, 2002.
- [21] G. Gnecco, "On the curse of dimensionality in the Ritz method," *J. Optim. Theory Appl.*, vol. 168, no. 2, pp. 488–509, 2016.
- [22] A. Alessandri, M. Gaggero, and R. Zoppoli, "Feedback optimal control of distributed parameter systems by using finite-dimensional approximation schemes," *IEEE Trans. Neural Networks and Learning Systems*, vol. 23, no. 6, pp. 984–996, 2012.
- [23] A. Quarteroni, *Numerical Models for Differential Problems*. Milan, Italy: Springer-Verlag, 2009.
- [24] Y. Iiguni, H. Sakai, and H. Tokumaru, "A real-time learning algorithm for a multilayered neural network based on the extended Kalman filter," *IEEE Trans. Signal Processing*, vol. 40, no. 4, pp. 959–966, 1992.
- [25] B. Schottky and D. Saad, "Statistical mechanics of EKF learning in neural networks," *Journal of Physics A: Mathematical and General*, vol. 32, no. 9, pp. 1605–1621, 1999.
- [26] K. Nishiyama and K. Suzuki, " H_∞ -learning of layered neural networks," *IEEE Trans. Neural Networks*, vol. 12, no. 6, pp. 1265–1277, 2001.
- [27] A. Alessandri, M. Cuneo, S. Pagnan, and M. Sanguineti, "A recursive algorithm for nonlinear least-squares problems," *Computational Optimization and Applications*, vol. 38, no. 2, pp. 195–216, 2007.
- [28] H. Mussa and R. Glen, "Memory-efficient fully coupled filtering approach for observational model building," *IEEE Trans. Neural Networks*, vol. 21, no. 4, pp. 680–686, 2010.
- [29] I. Yang and C. Tomlin, "Identification of surface tension in mean curvature flow," in *Proc. American Control Conf.*, 2013, pp. 3290–3295.
- [30] I. Yang and C. Tomlin, "Regularization-based identification for level set equations," in *Proc. IEEE Conf. on Decision and Control*, 2013, pp. 1058–1064.
- [31] A. Alessandri, P. Bagnerini, and M. Gaggero, "Extended Kalman filtering to design optimal controllers of fronts generated by level set methods," in *Proc. IEEE Conf. on Decision and Control*, 2016, pp. 3966–3971.
- [32] M. Falcone and R. Ferretti, *Semi-Lagrangian Approximation Schemes for Linear and Hamilton-Jacobi Equations*. SIAM, 2014.
- [33] A. Alessandri, P. Bagnerini, and M. Gaggero, "Optimal control of PDE-based systems by using a finite-dimensional approximation scheme," in *Proc. American Control Conf.*, 2013, pp. 1278–1283.
- [34] S. Fornaro, S. Maniglia, and G. Metafune, "Spazi di funzioni Hölderiane," in *Equazioni ellittiche del secondo ordine*, S. Fornaro, S. Maniglia, and G. Metafune, Eds. Edizioni del Grifo, 2004, vol. 4, pp. 51–68.
- [35] I. Capuzzo-Dolcetta and P.L. Lions, "Hamilton-Jacobi equations with state constraints," *Trans. Amer. Math. Soc.*, vol. 318, no. 2, pp. 643–683, 1990.
- [36] M.G. Crandall and P.L. Lions, "Hamilton-Jacobi equations in infinite dimensions I. Uniqueness of viscosity solutions," *J. Funct. An.*, vol. 62, no. 3, pp. 379–396, 1985.
- [37] M.G. Crandall and P.L. Lions, "Hamilton-Jacobi equations in infinite dimensions II. Existence of viscosity solutions," *J. Funct. An.*, vol. 65, no. 3, pp. 368–405, 1986.
- [38] A. Kurdila and M. Zabrankin, *Convex Functional Analysis*. Basel, Switzerland: Birkhäuser Verlag, 2005.
- [39] A. Barron, "Universal approximation bounds for superpositions of a sigmoidal function," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 930–945, 1993.
- [40] V. Kůrková and M. Sanguineti, "Comparison of worst-case errors in linear and neural-network approximation," *IEEE Trans. Inf. Theory*, vol. 48, no. 1, pp. 264–275, 2002.
- [41] A. Pinkus, *N-widths in Approximation Theory*. Springer Verlag, New York, 1986.
- [42] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Computation*, vol. 3, no. 2, pp. 246–257, 1991.
- [43] M. Leshno, V. Ya, A. Pinkus, and S. Schocken, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function," *Neural Networks*, vol. 6, no. 6, pp. 861–867, 1993.
- [44] D. Bertsekas, *Nonlinear Programming*. Athena Scientific, 2nd edition, Belmont, MA, 1999.
- [45] J. Humpherys and J. West, "Kalman filtering with Newton's method," *IEEE Control Systems*, vol. 30, no. 6, pp. 101–106, 2010.
- [46] K. Reif, S. Günter, E. Yaz, and R. Unbehauen, "Stochastic stability of the discrete-time extended Kalman filter," *IEEE Trans. Automatic Control*, vol. 44, no. 4, pp. 714–728, 1999.
- [47] I. Mitchell, "The flexible, extensible and efficient toolbox of level set methods," *J. Scientific Comput.*, vol. 35, no. 2-3, pp. 300–329, 2008.



Angelo Alessandri (S'88-M'97-SM'02) received the "Laurea" degree in electronics engineering and the Ph.D. degree in electronics engineering and computer science from the University of Genoa, Genoa, Italy, in 1992 and 1996, respectively. From 1996 to 2005, he was a Research Scientist with the National Research Council of Italy, Italy. In 2005, he joined the University of Genoa, where he is currently an Associate Professor. Dr. Alessandri was an Associate Editor of the IEEE Transactions on Control Systems Technology, the IEEE Transactions

on Neural Networks, and the IFAC Journal of Engineering Applications of Artificial Intelligence. He is an Editor of the International Journal of Adaptive Control and Signal Processing and an Associate Editor of Automatica.



Patrizia Bagnerini received the "Laurea" degree in applied mathematics from the University of Genoa, Genoa, Italy in 1994 and the Ph.D. degree in numerical analysis from the University of Nice, Nice, France, in 2002. From 2005 to 2015, she was a Researcher with the University of Genoa, and from 2016 she is Associate Professor at the Department of Mechanical Engineering of the University of Genoa. Her current research interests include numerical methods for hyperbolic problems, models for wound healing in biological tissues, and optimal control

problems. Her research works are mainly related to the approximation and control of problems for partial differential equations in different applications.



Mauro Gaggero received M.S. degree in electronics engineering and the Ph.D. degree in mathematical engineering from the University of Genoa, Genoa, Italy, in 2005 and 2010, respectively. Since 2011, he has been a Research Scientist with the National Research Council of Italy, Genoa, Italy. His current research interests include control, estimation, and optimization of nonlinear systems, distributed parameter systems, neural networks, and functional optimization problems. Dr. Gaggero is an Associate Editor of the European Control Association Conference Editorial Board and of the IEEE Control Systems Society Conference Editorial Board.